

Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation

Abdul Saleem, David Jazani, Hong Qing Yu

Institute for Research in Applicable Computing, University of Bedfordshire, Luton, UK

Email: Abdul.Saleem@study.beds.ac.uk, David.Jazani@beds.ac.uk, Hongqing.Yu@beds.ac.uk

How to cite this paper: Saleem, A., Jazani, D. and Yu, H.Q. (2019) Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation. *Open Journal of Energy Efficiency*, 8, 95-128. <https://doi.org/10.4236/ojee.2019.83007>

Received: June 6, 2019

Accepted: September 1, 2019

Published: September 4, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). <http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The Internet of Things has brought a vision to turn the digital object into smart devices by adding an intelligence system and thereafter connecting them to the internet world. These smart devices accumulate environmental information with the help of sensors and act consequently without human intervention. The Internet of Thing is a rapidly growing industry with expected 50 - 200 billion smart devices to connect to the internet. Multi-billions of smart devices will produce a substantial amount of data to provide services to human society, although, it will lead to increase energy consumption at the highest level and drive to high energy bills. Moreover, the flood of IoT devices may also lead to energy scarcity. IoT is nowadays mainly focused on the IT industry and researchers believe the next wave of IoT may connect 1 trillion sensors by 2025. Even if these sensors would have 10 years of battery life, it will still require 275 million batteries to be replaced every day. Therefore, it is a necessity to reduce energy consumption in smart devices. **“Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation”**, a proposed novel approach in this research paper by the help of a proposed algorithm in this research paper to reduce power consumption by individual devices within smart homes. In the proposed approach, a centralized automation controller keeps the less priority smart devices into deep sleep mode to save energy and experiments suggest the proposed system may help to reduce 25.81% of the energy consumed by smart devices within the smart home.

Keywords

Automation Controller, IoT, Energy Consumption, Arduino Uno R3

1. Introduction

Smart Home delivers comfortable living atmosphere by intelligent management system. Smart Homes become very popular research topics among Internet of Things (IoT) [1] [2]. Numerous IoT sensors connected through wireless communication medium within homes to make residence life comfortable [3] [4]. Architectural engineers of smart homes configure smart house before construction starts, so that, it helps to develop a well-designed low-cost smart homes. Furthermore, smart homes must have interaction system in between components and users to provide appropriate services, where components have sensors installed on it to detect various environmental data. Moreover, each sensor work independently and transfer data to controller, to get numerous environmental data. Multiple sensors must be placed within homes; however, multiple sensors around homes may increase installation cost [5]. It is expected in near future, more than 90 million people will be living in smart homes equipped with latest technologies for home automation to improve security, comfort and manage energy consumption. Several types of research have been taken place and suggest energy consumption can be reduced [6]. In smart home automation system devices incorporate in between devices to control home features, they do not only turn devices on/off [7] but also change devices internal behavior according to presence/absence of inhabitant. In result devices are being autonomously managed by itself without human intervention which also help to reduce energy cost as Electric consumption in Europe is rising continuously, especially in the residential sector and it is expected to grow approximately 50% by 2050. It is mainly caused by continuously increasing demands in electric plug-in-hybrid electric vehicle (PHEV), electric heating systems and others appliances such as Smart TVs, PCs, and smartphones which consumed nearly 25% of electric consumption within single homes in Europe. However, heating and cooling system of all buildings consumed approx. 65% of electric [8]. Smart homes contain intelligent home management system to provide the highest level of comfort to the inhabitant. It is one of the popular research topics in IoT (Internet of Things) [2]. It has a variety of IoT-based sensors installed in smart homes to improve inhabitant's life [3] [4]. Furthermore, smart homes constantly interact with components in order to facilitate appropriate services for inhabitants, sensors within smart homes work independently and simply report to controller [5].

In this research paper, a proposed work that have centralized approach an automation controller consists of several sensors and hardware on it such as ESP8266-ESP01 (a Wi-Fi module), BLE 4 (SH-HC-08), HC-SR501 PIR motion detector, LDR (Light Dependent Resistor), DHT 11 Humidity and Temperature, ACS712 to measure energy consumption and a very important component Arduino Uno R3 or a microcontroller that controls all sensors and hardware on it. Automation controller also has two loads on it and gets the behavior of the loads as per a proposed algorithm in this research paper called PA-PSM (Presence

Aware-Power Saving Mode). A proposed work in this research paper, HC-SR501 PIR motion detector plays a vital role as sense inhabitant presence within home. If the sensor found any activity, it informs the controller about it and controllers keeps the loads in switch on mode to keep inhabitant comfort at highest level. However, in absent of inhabitant within home PIR sensor sends the information to controller that no activity arounds so controller can send loads in to switch off mode to save energy. Furthermore, a proposed algorithm PA-PSM is also in placed which helps microcontroller to decide how to react in the absence and/or presence of human. Automation controller also has the ability to schedule the loads as per inhabitant requirement and PA-PSM helps automation controller to deal with the situation. In order to compare the results, two controllers were in placed automation controller and non-automation controller. These both controllers have similar loads on it. However, automation controller has additional PIR, LDR sensors and a proposed PA-PSM algorithm on it to work efficiently. After several experiments it is suggested that at least 25.81% of energy could be saved.

2. Literature Work

A smart home is equipped with sensors to collect occupant's activities, environmental changes and send data to controllers [9], reducing energy consumption can only be obtained by having profiles of appliance which shows consumption of individual appliances and help accordingly to change their behavior which is called Demand Side Management (DSM), it directly impact on consumer's behavior of electric consumption, moreover, it can shift some appliances such as dishwasher and washing machine usage from peak time to off-peak time to reduce cost of energy, so that energy optimization may help to smart house to reduce electricity consumption [10]. In another type of approach is Time of Use (ToU) pricing, in which price of electricity already set for specific period of time in advance, prices of electricity has already been paid for these period in advance, which allow users to shift their all energy usage to low cost period to reduce energy cost as it is implied on higher electricity demand with lower electricity prices instead of flat tariff, in another approach where a day is divided into several slots of with equal span with different electricity prices to minimize the cost of electric consumption [11].

Wireless Sensor Networks (WSNs) are playing a vital role in smart homes application and rapidly integrating into our daily lives, they are being used in several areas with numerous applications. WSN is being used in smart homes and offices to provide the best comfort to the inhabitant, as recently energy grids are being converted into the smart grid so consumers can directly control the consumption of their appliances, WSN is playing a vital role in energy management at the domestic side. Information and Communication Technology (ICT), energy harvesting, monitoring, smart meters with smart wireless devices become more important such as Wireless Sensor Home Area Network (WSHAN) [12], it

focused on communication in between smart Energy Management Unit (EMU) and smart appliances in nearly real time, and it can be managed better by another approach called BluHEMS.

BluHEMS is based on communication among central Energy Management Unit (EMU), home appliances, storage unit inside the smart home and smart meter. Home Energy Management (HEM) is suitable for smart homes to reduce energy cost by peak load management into off-peak hours can benefit smart homes. In BluHEMS communication system is based on Bluetooth Low Energy (BLE) protocol by WSHAN [13] as shows in **Figure 1**.

2.1. Smart Home System

Smart Home System consists of various energy sources, electrical home appliances, storage device, control unit and communication system, the home consist of own energy sources by renewable energy, for instance, solar panels and wind turbine, later these generated energies could store into batteries. Moreover, charging station within smart homes work as an inverter to convert direct current from renewable sources so home appliances can use it.

2.2. Wireless Sensor Home Area Network

Wireless Networks is great system for smart home applications as scalability of adding new devices within existing system is very easy, wireless links can be used to transfer both information such as power consumption as well as amount of time usage of appliances and smart plugs can be utilized to on/off devices as per load management, there are several Wireless Technologies, for instance, Bluetooth Low Energy (BLE) [13], ZigBee IEEE 802.15.4 [14] and Wi-Fi IEEE 802.11 [15]. These all have data transfer support, sensing and control system. These protocols can utilize existing system and that makes wireless system ideal for smart homes to manage energy consumption, where sensing nodes within smart homes are mostly battery powered, so low power feature is fundamental requirement for energy consumption of entire house, therefore it is required to choose best wireless protocol that is a key for energy savings. Bluetooth Low Energy (BLE) operates on 2.4 GHz ISM band with 40 channels, and capable of transferring 1 Mbit/s and the majority of the Bluetooth devices in the market today can support communication up to 10 meters in traditional Bluetooth but BLE may support up to 350 meters of range and beyond.

In the home sensor networks, longer range is required that may fit a normal size of homes as shown in **Figure 2**, composed of several independent Wireless Home Automation Cells (WHACs), controlled by BLE Master devices which act as Energy Management Unit (EMU) and participate with a smart meter and appliances connected with smart plugs.

2.3. Motivation for the Wireless Protocol Choice

The WSHAN's communication system based on Bluetooth Low Energy (BLE)

protocol [13], it is the lowest power consumption in Wireless Sensor Networks communication, low-cost sensors fit for the operation of indoor application [16], it has several reasons to choose BLE as in most of IoT applications, BLE is mostly the first choice because of low cost, low power and small devices for smart homes, unavailability of IEEE 802.15.4 is the main lack of mobile devices such as smartphones, tablet and laptop, moreover, IEEE 802.15.4 faces competition with BLE in terms of low cost and low energy, and Bluetooth consume less energy as compare to ZigBee, therefore, Bluetooth is the first choice for smart home automation.

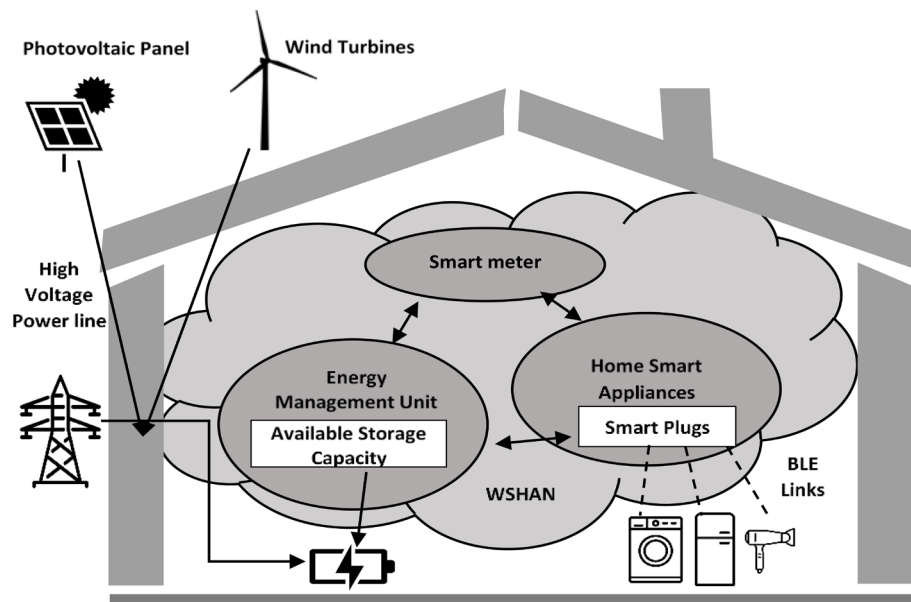


Figure 1. The BLE-based WSHAN topology.

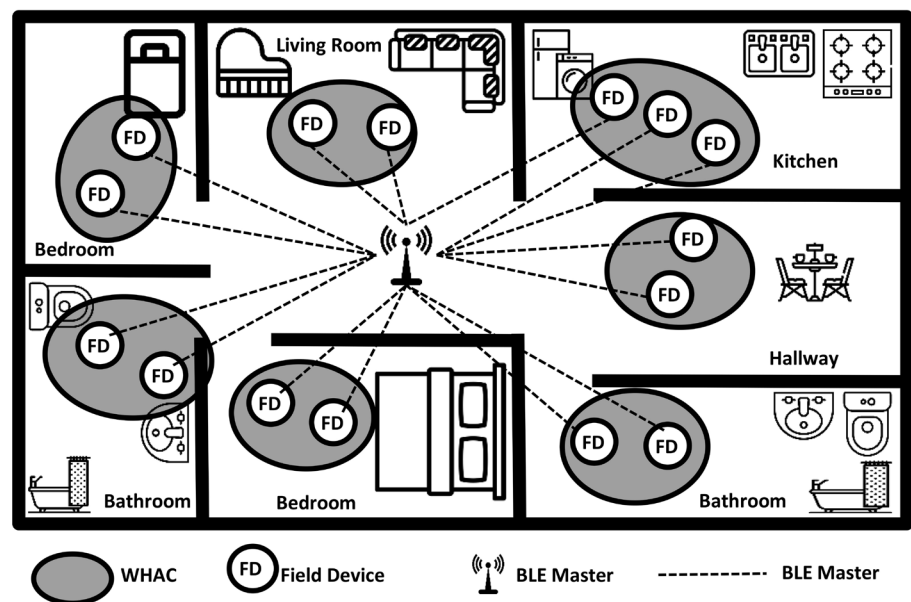


Figure 2. The BLE-based WSHAN topology.

2.4. BluHEMS Algorithm

BluHEMS to monitor and controls appliances within homes, in BluHEMS holds information of electric prices and energy storage systems to balance demand of electric supply to manage cost-effective operations, depicted diagram of BluHEMS shown in **Figure 3**, it has been developed in scenario mode, with an existing version of HEMS to comparison with, BluHEMS comparison considered by four appliances loads by, a coffee maker, a dishwasher a dryer and a washer, moreover, energy consumption of duration are vendor specific for appliances, average load were managed [17], above appliances consumed 0.3 kWh, 1.19 kWh, 2.46 kWh and 0.88 kWh respectively, with duration of cycles are 10, 30, and 60 minutes, furthermore, 4 kWh of extra loads have been assumed, therefore, 20% of loads are from standby appliances and rest from others appliances [17].

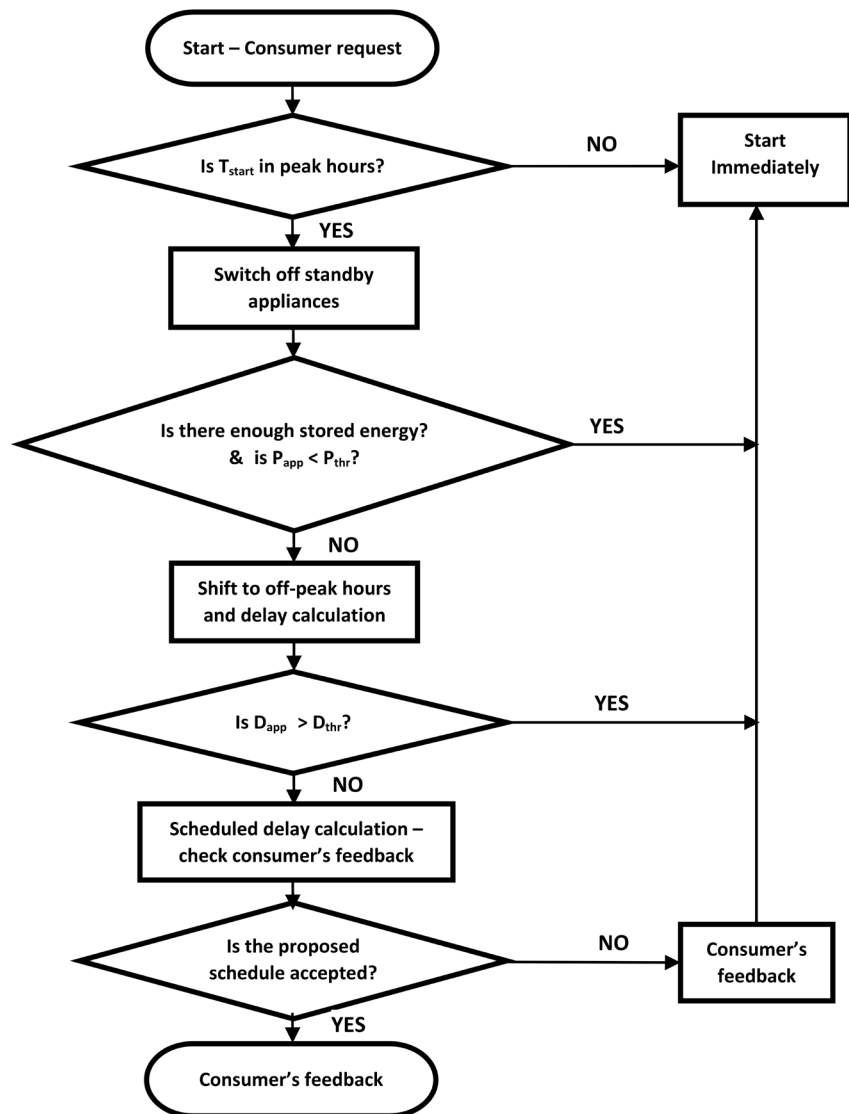


Figure 3. The BluHEMS algorithm chart.

In the next session, a research has been carried out to find out a comparative analysis in between BLE 4 and BLE 5, and found the BLE 5 has consumed 10% - 20% less energy in the same scenario, the reason of the comparative analysis is because BLE 5 based sensors weren't found during the experiments as BLE 5 was the latest release at that time and wasn't widely available, therefore Comparative analysis in between BLE 4 and BLE 5 conducted and chosen BLE 4 for the experiment results.

3. Comparative Analysis between BLE 4 and BLE 5

3.1. Overview

An indoor scenario to evaluate the performance of the latest version of BLE 5 as compared to the previous release of BLE 4 which is extensively utilized in every field in wireless communication. BLE 5 is an improved version of BLE with possible expect to make it more reliable in terms of less Power Consumption, wide range and higher data throughput which is already enabling BLE to rule on the Internet of Things to provide best solutions. The outcomes of the experiment are better communication range and throughput with less power consumption as it is being claimed since the release of the BLE 5 and it is achieved by using semiconductor Nordic nRF52840 chipsets. On the other hand BLE 4 is lower version of BLE broadly in used all over the world and found almost every devices such as laptop, tablets, smartphone and many other wireless devices, the recent version of BLE 5 has been released in early 2017, the latest version of the Samsung Galaxy 8 is the world first smartphone with the support of BLE 5. The feature of long-range has actually made BLE 5 highly adaptable of Internet of Things applications. Official announcement says BLE 5 has capability wide range of up to 4 times higher as compared with BLE 4 [18]. The results are led by using BLE 5 and BLE 4 chipset so it will help to find how BLE 5 is efficient in the application as compared to BLE 4 with the same application and environment.

3.2. Features of the BLE 5 Technology

The BLE 5 is a newer addition into the BLE family and addresses to comprehensive major problems to its predecessors have, such as limited range of communication, delaying in long transmission and broadcasting modes with limited functions. BLE 5 addresses the longer range of improved communication with maximum throughput by introducing new physical layer (PHY). BLE 4 has 1 Mbits/s Gaussian Frequency Shift Keying (GFSK) addressed in BLE 5 specifies 2 Mbit/s GFSK PHY called LE 1 M. The BLE 5 has 2 Mbit/s GFSK PHY called LE 2 M high-speed transmission for short-range communication however, 2 coded PHY with payload either 125 kbit/s or 500 kbit/s. the PHY layer LE coded using GFSK however, the payload is in two stages: first for error correction encoder and spread by pattern mapper. It enables to improve the link budget over 5 dB and 12 dB comparison with LE 1m and LE coded at 125 kbits and 400 kbit/s correspondingly.

Another big change to BLE 5 is the improvement of communication range with maximum transmission power of BLE from 10 dBm (10 mW) to 20 dBm (100 mW), unfortunately, some regions such as EU, Japan and Korea have imposed restriction on higher transmission power on BLE 5 therefore, higher transmit power does not provide any benefit to that region. BLE 4 has a maximum link layer PDU (Protocol Data Unit) was increased from 39 to 257 octets, which is stayed the same in BLE 5 as well. The co-existence of the devices uses 2.4 GHz has been addressed in BLE 5 by introducing special interface proving signals and messaging mechanisms.

Furthermore, broadcasting channels in BLE 5 has been tremendously improved by introducing an extended advertising feature. The format of advertising packets has been reworked in BLE 5, as BLE 4 was using 37 octets in primary channels however BLE 5 uses 255 octets of PDU as compare to the older version. Another feature enabled in BLE 5 which is the periodic advertisements, its broadcast the packets with intervals time ranging from 7.5 ms to 82 s so device synchronization with one or even many others several-overlapping advertisement, this feature of BLE 5 increase reliability and efficient solution of data broadcasting than any previous versions of Bluetooth, however both periodic and extended advertisement are options, more importantly, the BLE 5 has backward compatibility with all earlier version of BLE, although all the above features are not necessarily supported, nonetheless, but BLE 5 has the ability to increase communication range, throughput with new modes of operation [18] as BLE claimed to provide up to four times high range, 8 times higher broadcasting capacity and double the bandwidth as shown in **Table 1**.

3.3. Hardware & Software

In the experiment, the very first commercial chipset was used that has the support of BLE 5, the nRF52840 [19] manufactured by Nordic Semiconductor. In the system of the chip (SoC) has been integrated with 2.4 GHz transceiver with ARM Cortex-M4F microcontroller, with programmed of S140 v5.0 soft device which is a pre-compiled and built-in library of BLE protocol developed by Nordic Semiconductor.

3.4. Indoor Measurement

The measurement was conducted inside the building in one of the universities, where the scenario as shown in **Figure 4**, red dot represents the receiving node in the experiment. Blue dots represent RSSI and measured values of throughput for LE 1 M PHY mode using 0 dBm transmit power, however, green dots represent the RSSI and measured values of throughput when BLE 5 long range (125 kbps data mode), 0 dBm Tx power with $S = 8$ coding were in place. The BLE 5 long-range mode with 9 dBm power of transmission is represented by an orange dot. In the scenario, most of the walls are plaster boards except the bathrooms' which are made of the concrete wall represents by green lines. The bathrooms

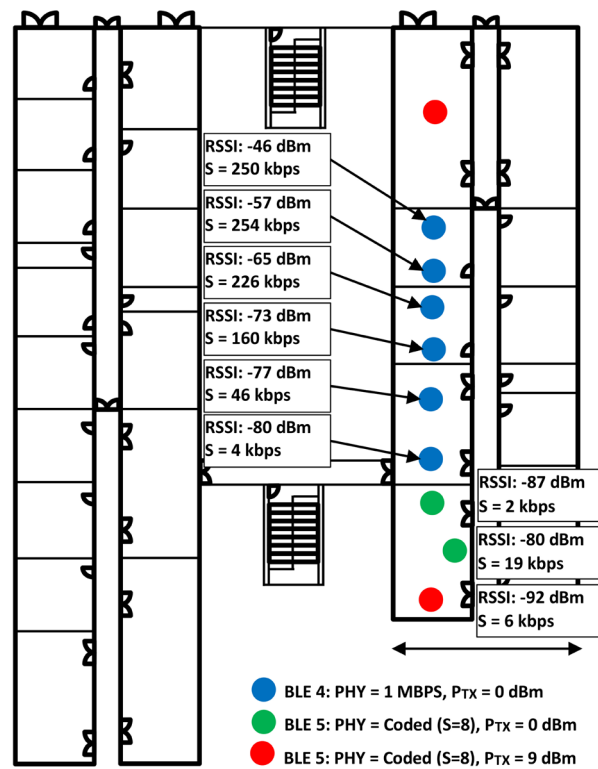


Figure 4. Results of the indoor measurements.

Table 1. Summary of BLE 5 specification.

PHY	Error Control	Range Multiplier	Packet Duration	Max Throughput
1 M	CRC	1×	80 μ s - 2.12 ms	800 kbps
2 M	CRC	0.8×	44 μ s - 1.064 ms	1438 kbps
Coded, S = 2	CRC & FEC	2×	462 μ s - 4.542 ms	382 kbps
Coded, S = 8	CRC & FEC	4×	720 μ s - 17.04 ms	112 kbps

and nearby surrounding have metallic pipers and fixtures, in experiments same devices were used for both plaster and concrete walls and experiment found that on the plaster attenuation was only 1 - 2 dB however through the concrete wall attenuation was 4 - 5 dB. The second scenario from fixed RX node on the fourth floor and TX node location from the third floor in this case attenuation measured in the building, in this first experiment, BLE 4 with transmit power 0 dBm and the RSSI was -13 dBm during TX and RX to close to each other. The RSSI value of receiver was -60 dBm with throughput 210 kbps, received value from floor 2nd was -81 dBm with throughput 40 kbps however floor 1 was not reachable by BLE 4, therefore, BLE 5 were used coded (S = 8) and then using 0 dBm power of transmission and still were not able to reach to the first floor, therefore, 9 dBm power of transmission and range with RSSI value -91 dBm and throughput up to 10 kbps. In this measurement with indoor improved commu-

nication has been achieved by using the coded BLE 5. Measurements between different floors such as from 4th to 2nd were achieved by BLE 4 however, to reach to the floor 1st by BLE 5 coded ($S = 8$) with extended transmission power. In this experiment, it has been found BLE 4 and BLE 5 with same 9 dBm transmission power and BLE 5 with 10% - 20% improvement in communication range. It is important as if BLE 4 and BLE 5 is in use for the same scenarios BLE 4 will consume more power to reach the distance of communication as compared to BLE 5. Therefore, BLE 5 will consume at least 10% - 20% less power to send and receive a message to the same scenario with BLE 4.

3.5. Discussion of the Results

BLE 5 has improved communication range as compared to BLE 4 which was the actual goal to achieve it. BLE 4 vs BLE 5 coded version compared and found BLE 5 is 10% - 20% less power consumption in the same indoor scenario with the same distance of transmitter and receivers, BLE 4 and BLE 5 both using 9 dB and it has found that BLE 5 has been found improvement of 10% - 20%.

In term to test proposed work of “**Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation**”, Proteus 8.6 was selected for the simulation but there are some limitations and challenges faced in Proteus 8.6 as it doesn't have the latest version of Bluetooth Low Energy in built-in library, therefore, practical work could not have been achieved with Proteus 8.6, therefore, the only solution the test rig was left, however, BLE 5 was the newest technology at that time, so that sensors weren't available in UK market in early 2017 as it was just the released at that time, however BLE 5 is not available yet compatible with Arduino Uno R3. These were the limitation and challenges; so, the test rig has been chosen and BLE 4 was utilized instead of BLE 5. It shows the BLE 5 is 10% - 20% better than BLE 4 in the same scenario. In the next section, the discussion carried out of challenges faced, implementation of the test rig and performance evaluations.

4. Challenges, Implementation & Performance Evaluation

4.1. Reason for the Test Rig

MATLAB Simulink [20] was the first option to develop the artefact, however due to complexity and unavailability of the components within built-in libraries become issues to drop the Simulink, complexity of the Simulink was also the main reason, on the other hand, module unavailability of Bluetooth Low Energy 5.0 at the time of experiments, it was brand new technology which wasn't available widely in the market, after dropping the Simulink, it was decided to go with other option. There were several options to go with for instance Autodesk Eagle [21], express PCB [22], and Proteus [23] developed by lab center. These all are top ten software tools by Electronics Lovers [24] to design PCB, however, Proteus was chosen to complete the task, and several past version of Proteus were tried, however, Proteus 8.6 finally chosen to complete the task. There were sev-

eral experiments conducted through Proteus on each module as required by the artefact, and it has also found limitations as given below:

- The latest version of Proteus doesn't have a most recent version of BLE 5 (Bluetooth Low Energy) module within their built-in libraries; it doesn't even have BLE 4 which is one of the oldest versions of BLE.
- ESP8266 module or Wi-Fi module was not found in Proteus as well, it is another important module after BLE for the experiments.
- Rest of other modules are available in Proteus but with some limitation such as it cannot give real data from the environment as sensors within Proteus doesn't sense the real environment.

4.2. Non-Automation & Automation Block Diagram

In order to determine energy savings in Presence Aware Power Saving Mode, first step to bring smart home appliances on centralized board so, these appliances could easily be controlled through microcontroller called Arduino Uno Rev3. In order to carry test, two sets of entire similar boards have been developed which are called non-automation and automation, these boards have Arduino Uno Rev3 on board along with ESP8266-ESP01 (a Wi-Fi module), BLE 4.0 (SH-HC-08), Load 1, Load 2 and three sets of current sensor called ACS712, however, automation board has some extra features including modules called HC-SR501 PIR Motion Detector, LDR (Light Dependent Resistor) and DHT 11 Humidity & Temperature, and a proposed algorithm on top of on automation board makes it work as “**Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation**”.

4.3. Non-Automation Block Diagram

In order to complete the task, block diagram for the non-automation designed first, it was designed to take comparative analysis task to next level, as if there was only automation board on then there would not have anything to compare with, therefore, two PCB boards were designed side by side with same functionality with only difference on automation board. A block diagram of non-automation is shown in **Figure 5**.

In the non-automation board, Arduino Uno R3 plays the major role therefore, placed on the middle of the PCB and rest of the modules around it, all modules are connected to Arduino Uno R3 directly. The power is only supplied to the main PCB board at first and then it has energy meter connect to sense what total power is being consumed by the PCB board including all modules on it. There are two more energy meter connects to ESP8266-ESP01 (a Wi-Fi module) and connects to BLE 4.0 SH-HC-08. These three energy meters are connected to monitor energy consumption for BLE 4 SH-HC-08, ESP8266-ESP01 and for the whole PCB including Arduino Uno R3. There are 2 LEDs on board could be switched on/off by given commands through a mobile application connects to SH-HC-08, however, Wi-Fi module is connected to Free IoT platform where it

sends the current sensor information as well as the status of the loads. Centralized approach has been used to connect home smart appliances to one page which is Arduino Uno R3 and thereafter, it controls home smart appliances centrally.

4.4. Automation Block Diagram

Automation block diagram looks a like the non-automation as shows in **Figure 6**, but it has extra modules on it with the addition of proposed algorithm which makes automation board work as “**Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices For Energy Conservation**”. Automation board also consists of Arduino Uno R3 on the middle of the board and additional sensors/modules around and connects to it. It also connects to the main power supply, which goes first to Arduino Uno R3 thereafter, it supplies to relevant modules/sensors as per the time they need to wake up based on the proposed algorithm. It is the best approach to get the smart home appliance to connect to the centralized microcontroller and switch them on/off as per their role or if schedule defined to turn the device on. In automation board, SH-HC-08 and ESP8266 have a similar role but additional sensor/module makes it smarter.

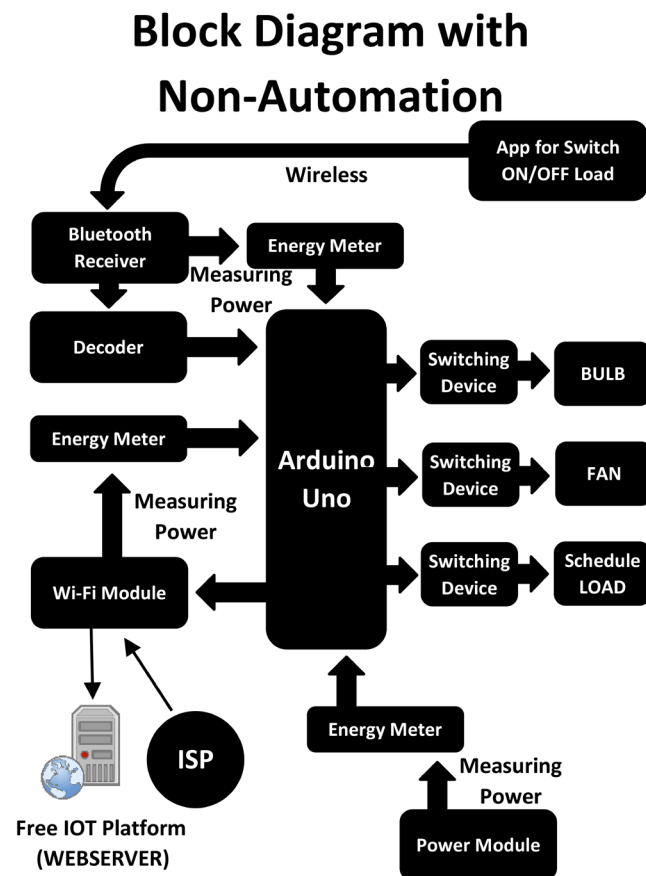


Figure 5. Block diagram for non-automation.

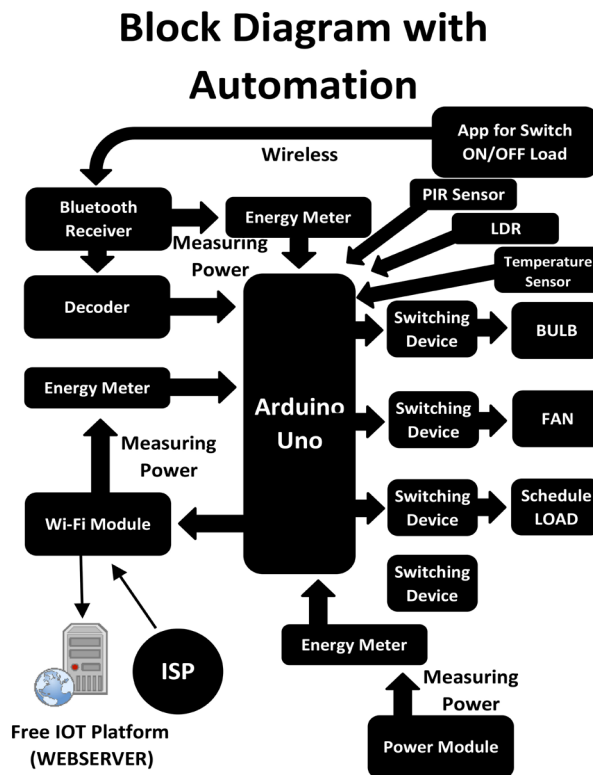


Figure 6. Automation Block diagram.

It has HC-SR501 PIR Motion Detector, Light Dependent Resistor, DHT 11 Humidity and Temperature Detector. PIR motion detector is used to detect motion around, in automation system. If there is a motion detected around means resident is home therefore, Arduino Uno R3 keeps home appliances ready to use as at any point resident may use it, on the other hand, if there is no motion detected around Arduino sends home smart appliances in switched off mode because no resident home means no smart home appliances to go switch on. LDR or Light Dependent Resistor to detect if it is required to switch lights on/off, during the daylight if there is enough light in the room then light bulb would be kept the switch off until residents intervene directly or through Bluetooth mobile app and want to turn the light on even if doesn't require. DHT 11 Humidity and Temperature detector which helps Arduino Uno R3 to decide to turn Fan or other cooling devices on/off, however, no Fan is connected physically at the moment on the board, finally RTC (Real Time Clock) helps Arduino to get the real-time thereafter, check if there is any schedule has been set up by resident for any particular home appliances, if done so, Arduino Uno R3 turn schedule load on until it finishes its task and turn off itself unless no schedule found to turn the device off.

4.5. Proposed Algorithm Flowchart for Automation

A proposed algorithm for the artefact, to aid Arduino Uno R3 to decide when and what situation loads could be sent to switch on/off mode. It is the algorithm

that plays the most important part to develop “**Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation**”. It is also called automation algorithm which defines the rules and written programming codes accordingly for the Arduino Uno R3 so that it can communicate with the sensors/modules on the board and decide to react accordingly, a flow diagram of automation algorithm given in **Figure 7**.

Once the programming codes start working on the board, it initializes sensors/modules first, thereafter, the Arduino Uno R3 checks if there is any device scheduled to run at specific time, if the time is due for any scheduled device, Arduino Uno R3 power it up to scheduled device, then continue the loop to go to PIR motion detector and it starts detecting the motion around, this cycle continues until any motion detected, on the other hand, if there is any scheduled device is on either it will finish its running cycle on-off itself or another schedule to turn schedule device off or resident either switch device off manually or by sending commands from mobile phone which it connects through Bluetooth, moreover, motion detectors cycle will continue to detect any motion around, once detects it sends the alert to Arduino, LDR or Light Dependent Resistor detects whether the light is required, if so then Arduino Uno R3 switch the load 1 on. Automation algorithm performs the task from several perspectives, for instance, if motion detected it can power up the loads, if no motion detected but resident connects to it through Bluetooth from distant without being detected by PIR, automation algorithm still works and gets the command from residents mobile apps connect via Bluetooth, even if there is no motion detection, no connection with Bluetooth to receive commands but resident has scheduled any device to switch on at a particular time and switch back to off after completing the running cycle, this automation algorithm still performs their task very well in all above scenarios. Since automation board has ESP8266-ESP01 Wi-Fi on board, a future recommendation can be added then these devices could easily be controlled through web or any mobile app connects through the internet and accessing the automation board to control home appliances, controlling home appliances through internet is future proposed work which is easily achievable as ESP8266-ESP01 is already on board.

4.6. Testing the Sensors/Operators

After selection of modules, sensors and setting up automation algorithm, in next step all sensors/modules have been tested in simulation mode through Proteus, it was tested basic functionality, connecting the module to Arduino, control through it. Once all available module/sensor tested in simulation mode next step was to design a virtual PCB (Printed Circuit Board) were designed in Proteus, it allows to draw PCB on it with the help of built-in components, draws the path or connection in between modules/sensors and later tests it that all paths or connection in between modules and sensors are correctly defined as per the electronics laws. There is 3D view the PCB designed in Proteus including PCB layout as given in **Figure 8**.

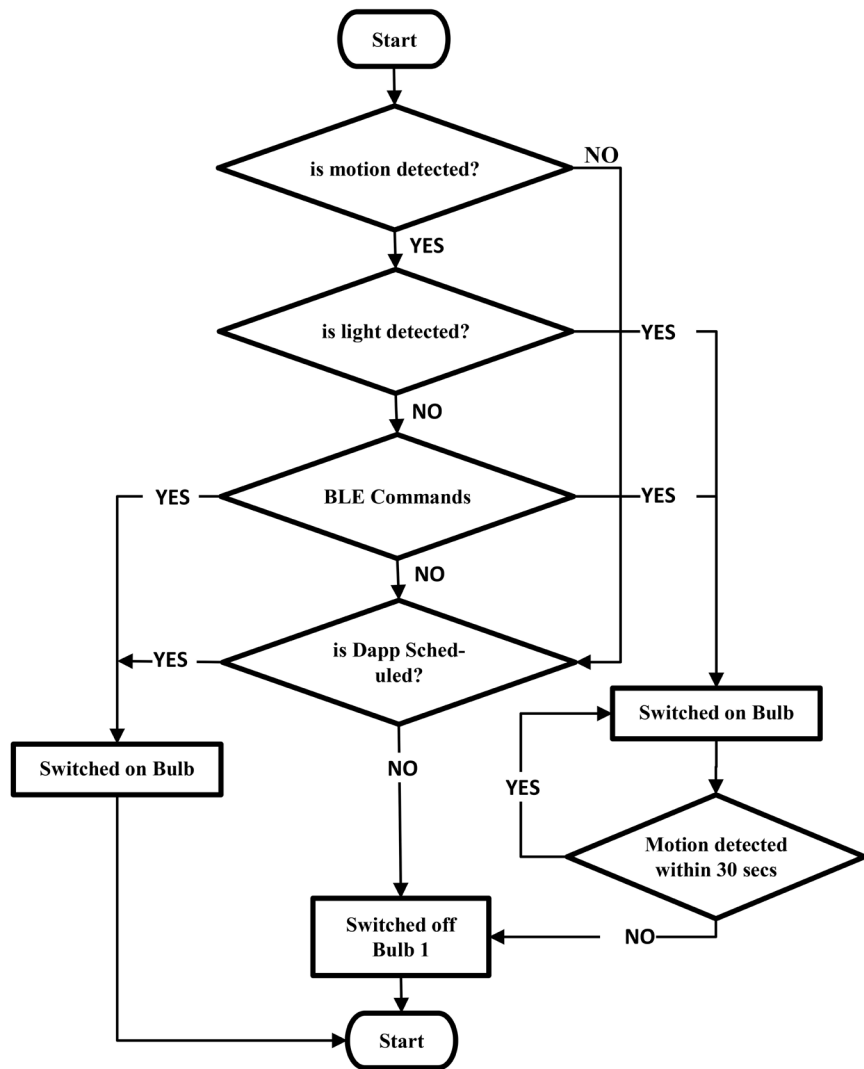


Figure 7. PA-PSM algorithm.

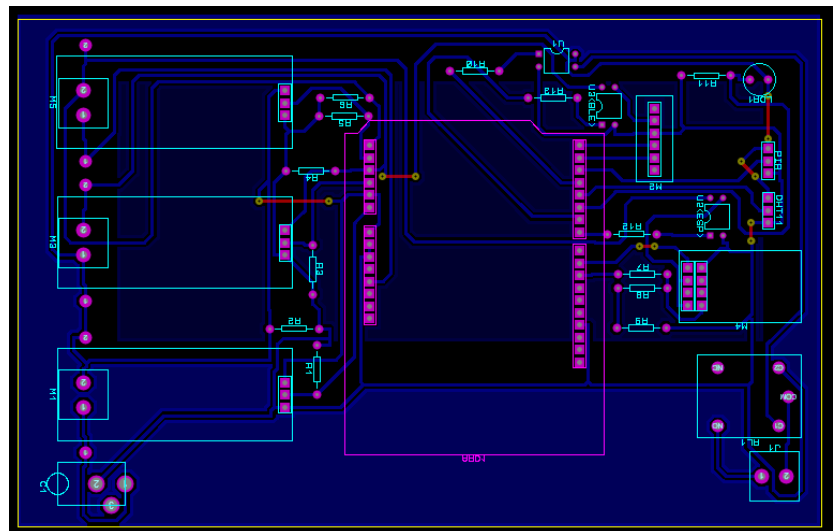


Figure 8. Soft version of printed circuit board.

PCB layout is shown which represents the position of the modules and sensors on the PCB, their paths or connection to the Arduino Uno R3 microcontroller on the board represent by pink box, in **Figure 9**, a 3D view of the printed circuit board.

In order to test hardware in physical mode, a PCB printed out first as shown in **Figure 9** thereafter, each operator placed on it and tested through the Arduino controller. In **Figure 9**, a red rectangle box, left bottom corner is a DC power supply to the main board, and just above it there are three rectangle green boxes given represents to ACS712, first ACS712 sense the current for the SH-HC-08 BLE, second is for ESP8266-ESP01 Wi-Fi module and third ACS712 for the whole PCB. Arduino has been placed at the middle of the PCB and just right top of the Arduino SH-HC-08 BLE has been placed, PIR motion detector and DHT11 Humidity and Temperature detector just adjacent to BLE and in the middle of the right-hand side a red rectangle box represents ESP8266-ESP01. These objects are connected to Arduino through a copper wire placed on the PCB board.

In the very first step, Arduino places on the board, and test with smallest programming codes by printing “Hello World!” to see the results on Serial Monitor provided by Arduino IDE. In the next step two LEDs place on the board and turn them on/off through Arduino, thereafter, SH-HC-08 place on the board, check their connectivity through the mobile application once passes, Arduino Uno R3 and SH-HC-08 communication take places. Now mobile application sends on/off command to BLE and BLE sends to Arduino to what bulbs needs to on or off as per given commands by SH-HC-08. in the next stage, PIR motion detector places on the board with the lowest sensitivity, if PIR detects any motion around, it sends the data to Arduino Uno R3 and thereafter it turns the LED on as motion detected else to keep the LEDs in switch off mode. DHT11 Humidity and Temperature detects the values and sends to Arduino Uno R3 to calculate if the threshold is hit to turn the fan or cooling device switch on/off. In the final stage, all programming codes combine into a grand code including automation algorithm programming codes and turn into one program so all modules/sensors can work as a team and work as automation board.

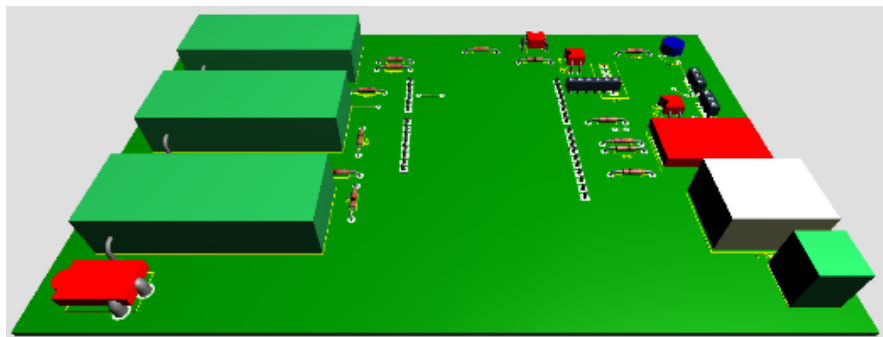


Figure 9. Printed circuit board in 3D.

4.7. Analysis of Operators

There were several experiments were carried out, however, three experiments have been listing and carried out comparative analyses in between non-automation and automation. The module and sensors attributes and settings were left with default values; however, baud rates for ESP8266-ESP01 and BLE SH-HC-08 modules were set to 9600 bps (Bit per Second). The artefact is developed on domestic scenario based for both non-automation and automation. as per researcher there would a flood of devices in the near future and each home may have up to seven IoT devices [25], although, two loads have been shown on each automation and non-automation controllers, these are also called smart controllers or smart communication systems, which will bring smart home appliances on board and control them, both controllers have similarities, however automation controllers have some extra features which make it differ from non-automation controller, automation controller have HC-SR501 PIR Motion Detector which detects human presence, if found then it keeps smart home appliances ready to use else automation controller keeps smart home appliances switch off. Automation controller takes up to 30 seconds to initialize all sensors especially HC-SR501 PIR Motion Detector to get ready to work as compared to the non-automation controller. nowadays, every electronics gadget built-in with Wi-Fi and now smart home appliances are also coming with built-in Wi-Fi because of IoT, therefore automation controller may keep less priority IoT devices on switch off mode if no resident home to save energy. However, automation controller also has scheduling system which means the smart home appliance will be switched-on at the scheduled time even if at non-resident home and switched back to off at given schedule time or itself switched off once the task completed. This approach can keep maximum modules of Wi-Fi off mode and save energy unless it is not priority device. In order to run the test rig and get graphs and tabulating data, non-automation and automation controllers both have started run at the same time and stopped at the same time with the same scenario to check how both responded and what energy they have consumed and what percentage of energy could have saved in terms to keep the loads switch on/off. There were several experiments recorded, many of them with the shortest period to cross-check if the devices heating up or any other technical issues, once modules and sensors have shown reliability thereafter it was run for the longest time and chosen data from three experiments.

5. Experiments & Results

5.1. Experiment 1

Experiment 1 was set up during the day time with starts time at 08:22 and it was run until 17:07 approximate 8 hours and 44 minutes of running time for both non-automation and automation controller, it was left in the common room so automation controller may have some motion detection as well, in **Table 2**, data shows consumption of non-automation controller. in the table it shows Total

watts hours, SH-HC-08 watts hours and the last column for ESP8266 watts hours, however number of watts consumption as per time is also given in to the first column of the data.

In 8.74 hours, it was noticed that total 47.89 watts of energy has been consumed by the non-automation board where more than half of the energy consumed by ESP8266, it is nearly 54% of total energy consumed by ESP8266 which is 9% extra energy consumption as compare SH-HC-08. In smart home appliances, nowadays these appliances are IoT based therefore, those appliances built-in with Wi-Fi and all connect to the internet all the time and consume energy. Non-automation board has two loads on it, which could be switched off/on by sending commands through a mobile application. Mobile connects to the non-automation controller through BLE. There is another table shows the data of loads it shows how often loads were kept on by sending commands from mobile.

Table 3 shows the duration of Load 1 and Load 2 were switched on for the total running time of non-automation controller. The controller running time was 8 hours and 44 minutes, however Load 1 was switched on for 1 hour and nearly 21 minutes and load 2 was switched on for 1 hour and nearly 20 minutes. Non-automation controller completely relies on user's interaction to get the loads to switch on/off by sending a command through Bluetooth via the mobile application, a table represents that load 1 was remained switch on 15.42% of the total running time of controller however load 2 was remained switch on for 15.25%. The loads on controllers are actually LEDs those represent the loads and it was assumed each LED is 100 w bulb therefore, if both loads of 100 watts were remained on for the full duration of controller running time it would have consumed 1748 watts of energy although both loads were remained switch on for 15.42% and 15.25% respectively so energy consumption for both were 134.8 watts for load 1 and 133.30 watts for load 2.

Automation controller, as discussed earlier have additional modules and sensors to make automation controller to keep smart home appliances switch off unless no residents are home, a table of energy consumed by the modules and sensors in automation controller given in **Table 4**. In the automation controller, 37.51 watts of energy consumed in 8.74 hours as compared with 47.89 watts consumed by non-automation control. It is found that the automation controller consumed 21.67% less energy, SH-HC-08 consumed 40.94% less consumption and ESP8266 consumed 29.54% less energy respectively.

In **Table 5**, belongs to the loads from automation table, load 1 in automation controller switch on only if resident is home or residents connect to the BLE without being detected by motion detector and send switch on/off commands through it, however, load 2 can either be switched on by given commands by BLE or the device is already scheduled to switch on. Load 1 in automation was switched on for only 10 mins and 26 seconds which is just 1.99% of total 8.74 Hours running time.

Table 2. Experiment 1—non-automation watts consumption.

Run Time	Total Wh	SH-HC-08 Wh	ESP8266 Wh
10 Mins	1.25	0.57	0.67
30 Mins	3.4	1.54	1.83
60 Mins	7.26	3.29	3.9
3 Hours	17.45	7.88	9.4
6 Hours	33.34	15.03	17.98
8.74 Hours	47.89	21.59	25.83

Table 3. Experiment 1—non-automation loads.

Run Time	Total Wh	SH-HC-08 Wh
10 Mins	1.25	0.57
30 Mins	3.4	1.54
60 Mins	7.26	3.29
3 Hours	17.45	7.88
6 Hours	33.34	15.03
8.74 Hours	47.89	21.59

Table 4. Experiment 1—automation watts consumption.

Running Time	Total Wh	SH-HC-08 Wh	ESP8266 Wh
10 Mins	0.77	0.26	0.37
30 Mins	2.27	0.77	1.10
60 Mins	4.44	1.50	2.16
3 Hours	13.36	4.53	6.49
6 Hours	25.73	8.73	12.50
8.74 Hours	37.51	12.75	18.20

Table 5. Experiment 1—automation loads.

Running Time	Load 1	Load 2
10 Mins	00:01:59	00:00:00
30 Mins	00:02:38	00:00:00
60 Mins	00:03:16	00:00:00
3 Hours	00:05:35	00:00:00
6 Hours	00:07:16	00:00:00
8.74 Hours	00:10:26	01:14:22

5.1.1. Main Watts Hours

A comparison graph of non-automation and automation controller given in **Figure 10**, in which x-axis represents the time and y-axis represents the watts have been consumed, the difference is very wide and clear as automation boards

have consumed 21.67% less energy than non-Automation. In the start of the device running time, the difference is not big in the start but at the end of the graph difference reach up to 10.37 watts which is equal to 21.67%.

5.1.2. SH-HC-08 BLE Watts Hours

“Bluetooth Low Energy Watt Hours” is another comparison graph taken from both non-automation and automation controller, in **Figure 11**, it can be seen BLE from automation have consumed less energy as compared to the other board, it has consumed 12.75 watts in total duration and that is 8.84 watts less or 40.94% less than the BLE consumed in the non-automation board.

5.1.3. ESP8266 Wi-Fi Watts Hours

ESP8266 Wi-Fi a comparison graph of watts hours is given in **Figure 12**, the graph looks like the BLE comparison graph, but it has noticed a gap in between is a little closer as compare to BLE. In order to stay connects with the internet, Wi-Fi was remained switch on most of the time. Therefore energy consumption by ESP8266 was at the highest level amongst all other modules and sensors. Energy consumption in automation ESP8266 was up to 18.2 watts during 8.74 Hours of running time which is 29.54% less consumption as compare to non-automation ESP8266.

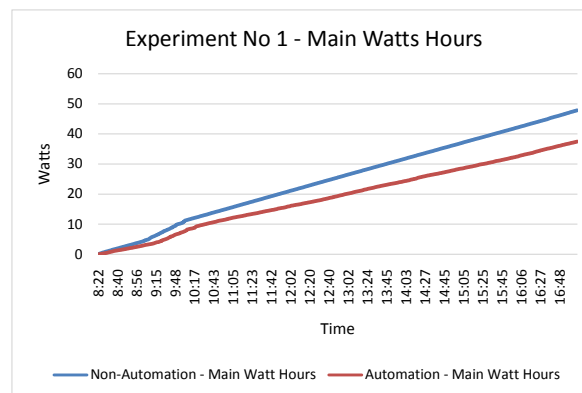


Figure 10. Main watt hours.

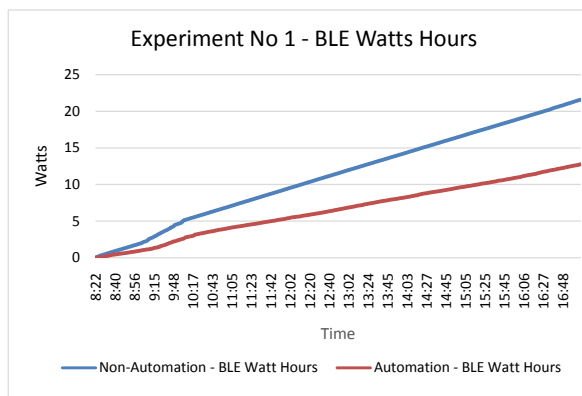


Figure 11. Bluetooth low energy watt hours.

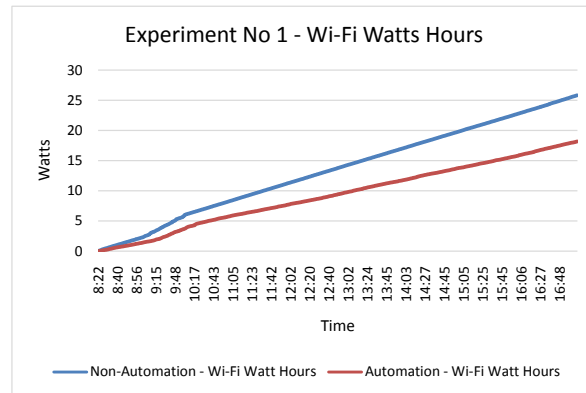


Figure 12. Wi-Fi watt hours.

5.1.4. Load 1 & Load 2

It has already explained, non-automation loads can only be switched on/off through BLE user commands, but in automation controller loads could be switched on/off by various method such as load 1 could be utilized either through motion detector or through BLE command, however load 2 cannot be switched on/off by motion detector but through scheduling if in case of the resident has set up. In experiment 1 show some improvement if having a look at both graphs line, as loads on automation controller both were switched on mode for 5 occasions only however, loads within non-automation were both switched on 183 times where the big chunk of both loads was switched on mode can be seen on the right-hand side of the graph as shown in **Figure 13**. In an average of both loads were switched on for 15.33% of average time however in automation controllers both loads were switched on 8.85% in 8.74 hours' time which is 6.48% less than the other, although the only load 1 is working by motion detector if both loads were controlled through motion detector than average consumption would have been 1.99%, therefore, total energy savings could have gone higher.

5.2. Experiment 2

The experiment 2 has lengthier than the last experiment, this experiment started at 06:10:33AM and lasted for 11.80 hours which is the longest running time for any experiments, it was for both non-automation and automation controller, this time both controllers left in the TV lounge and still lots of motions were detected. A tabulated data is given in **Table 6** belongs to the non-automation controller, tables show the same types of data as shown in Experiment 1, the first column shows the running time of the controller, second column non-automation controller's watt-hours and a third column for the SH-HC-08/BLE watts and in the last column shows the ESP8266/Wi-Fi watts consumed.

In 11.80 hours of time, non-automation board's energy consumption reached to the 56.08 watts and ESP8266 once again consumed more than half of total

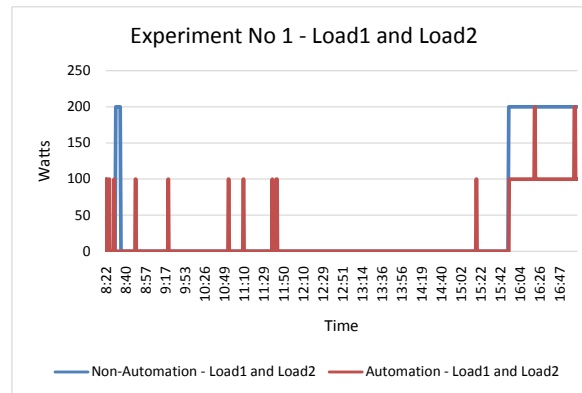


Figure 13. Experiment 1—Load 1 and Load 2.

Table 6. Experiment 2—non-automation watts consumption.

Running Time	Total Wh	SH-HC 08 Wh	ESP8266 Wh
10 Mins	1.03	0.47	0.55
30 Mins	3.1	1.4	1.67
60 Mins	5.87	2.64	3.17
3 Hours	15.75	7.07	8.52
6 Hours	30.35	13.21	16.84
11.80 Hours	56.08	23.01	32.51

board energy, it has consumed 57.97% of the total energy, which is 16.94% more than SH-HC-08 BLE, in recent days IoT devices gain more attention than any other I.T. branches therefore, Wi-Fi modules are much common in smart home appliances, in all experiments it was noticed that ESP8266 consumes more energy than any other module have utilized in this artefact, this experiment is as similar as Experiment 1 the only difference is with this time controller running time exceeded additional 3.06 hours, both loads on non-automation could be controlled by mobile application through BLE connectivity.

Table 7 shows the data of Load 1 and Load 2 in Experiment 2 was decided to keep all loads to the working load and the capacity of the controllers. It has shown that the Load 1 was switched on for 11.55 hours and Load 11.54 hours, however, total running time of the controller was 11.80 hours, and thus both loads were kept switch on for the average of 97.88% of the total running time of the controller. Non-automation controller's load could only be controlled by the user's commands given by mobile application connected to the controller through SH-HC-08 BLE.

Automation tabulated data, as shown in **Table 8**, was kept on for a similar duration of 11.80 hours, automation controller consumed a total of 49.81 watts of energy which is 11.20% less consumption than the non-automation controller. ESP8266 has performed well here in terms of energy consumption and it has consumed 4% less energy as per compare to non-automation controller's ESP8266,

however both modules in each controller have the same loads, Load 1 in automation mostly based on motion detector therefore, ESP8266 has 36% of less data to handle through Wi-Fi. ESP8266 could have been performed better if both loads were switched off at the same time as there would not have anything to deal with ESP8266 which help to reduce energy consumption.

Automation **Table 9** belongs to the Loads is given, as it is given Load 2 was switched on for time of 11:32:17 which is 97.78% of the total running time of the controllers, however, Load 1 was switched on for the duration of 02:51:38 which is 24.24% of the total running time, average of both loads were 61.01% of total running time which is 36.87% or less time as compare to non-automation loads. in the start of the automation controller's both loads have a similar duration of running time, however, the differences were starting to reflect after half an hour where load 1 kept off due to in the absence of residents.

Table 7. Experiment 2—non-automation loads.

Running Time	Load 1	Load 2
10 Mins	00:07:39	00:06:44
30 Mins	00:27:38	00:26:43
60 Mins	00:58:02	00:57:07
3 Hours	02:53:48	02:53:31
6 Hours	05:54:07	05:53:31
11.80 Hours	11:33:06	11:32:50

Table 8. Experiment 2—automation watts consumption.

Running Time	Total Wh	SH-HC-08 Wh	ESP8266 Wh
10 Mins	0.737	0.27	0.39
30 Mins	2.277	0.85	1.2
60 Mins	4.587	1.72	2.41
3 Hours	13.86	5.19	7.29
6 Hours	26.818	9.99	14.15
11.80 Hours	49.808	18.45	26.38

Table 9. Experiment 2—automation loads.

Running Time	Load 1	Load 2
10 Mins	00:06:17	00:06:44
30 Mins	00:14:08	00:26:43
60 Mins	00:28:35	00:57:07
3 Hours	01:29:24	02:54:19
6 Hours	02:27:40	05:52:12
11.80 Hours	02:51:38	11:32:17

5.2.1. Main Watts Hours

A comparison graph as given in **Figure 14**, non-automation and automation controller from Experiment 2, in the comparative analysis, automation controller has consumed 11.18% less energy than non-automation controller, the difference was minimum at the first few minutes thereafter, difference was gradually improved and automation controller consumed 49.80 watts during the whole duration of 11.80 hours, however non-automation consumed 56.08 watts. It was the longest running time for any experiments and have seen 11.18% less consumed by the automation controller.

5.2.2. SH-HC-08 BLE Watts Hours

There is another graph given in **Figure 15**, a comparison in between BLE of non-automation and automation controller, 19.82% less energy consumed by automation controller during the same time as non-automation however, it consumed 23.01 watts of energy as compared to just 18.45 watts by the automation controller, which is 4.56 watts less than BLE from non-automation controller, as it can be seen in the start it was slightly difference but it kept gradually increased up 19.82%.

5.2.3. ESP8266 Wi-Fi Watts Hours

ESP8266 ESP-01 Wi-Fi, a comparison graph from non-automation and automation controller given in **Figure 16**, the 6.13 watts of difference has been noticed in between. Moreover, 32.51 watts of energy consumed by non-automation and 26.38 watts consumed by ESP8266 in automation controller which is 18.86% less consumption. ESP8266 consumption remains higher as most of the time it stays connected to maintain quality of service.

5.2.4. Load 1 & Load 2

Load 1 and Load 2, a comparison graph in between shows in **Figure 17**, as it can be seen at the start of the loads and until 10:55 both loads on both controllers were remain switched on most of the time, and some part of the end of the graph shows nearly half of the time both loads on both controllers were on. However, big chunk in between 10:56 to 15:15, automation board has only load 2 was switched on which is based on scheduling, but load 1 which is based on motion detection was switched off in between 10:56 to 15:15 due to unavailability of residents, that big chunk actually help to reduce dramatically. In non-automation both loads were on 1602 times however in automation it was only for 380 times, it is even less than a quarter of the non-automation loads.

5.3. Experiment 3

Experiment 3 and the final experiment of the artefact were conducted during the night time to find out the behavior of both microcontrollers, Experiment 3 was started from 19:57:19 and ended up by 05:59:34 in the morning. Moreover, controllers' total running time was 10.02 hours which is less than experiment 2 although more than experiment 1. Experiment 3 was set up in the kitchen room

where all smart home appliances usually exist including a fridge and a washing machine. A tabulated data is shown in **Table 10** that shows consumption of non-automation controller along with SH-HC-08 as well as ESP8266.

In 10.02 hours of running time, it was 50.73 watts of total energy consumed by non-automation board where 51% of the energy only consumed by ESP8266 up to 25.82 watts. However performance of SH-HC-08 was better than ESP8266, it has consumed 17.85 watts of energy in total running time of non-automation controller which is 16% lower as compared to ESP8266.

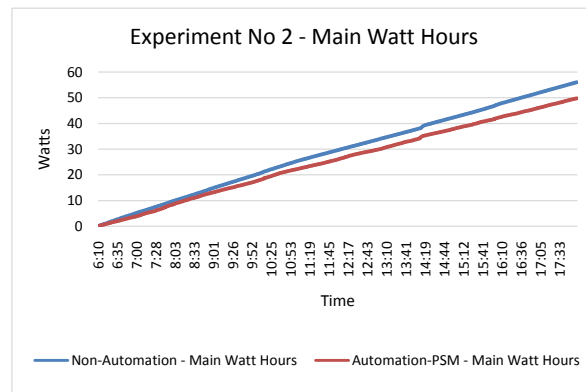


Figure 14. Main watt hours.

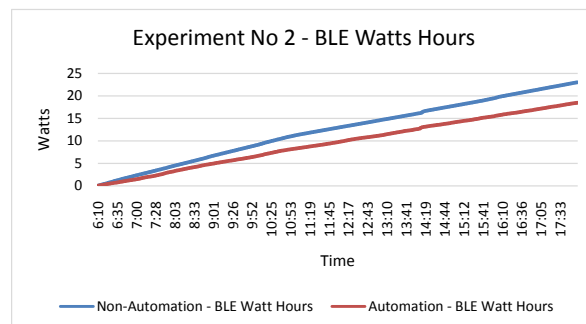


Figure 15. Bluetooth low energy watt hours.

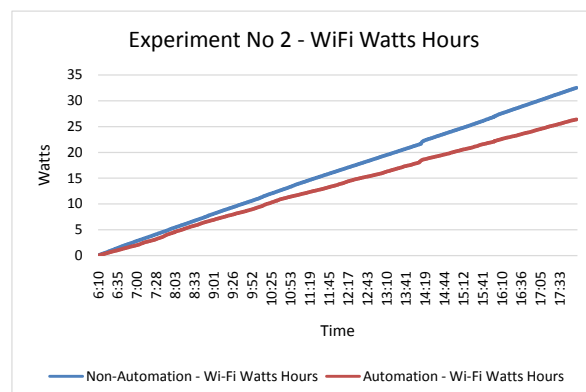


Figure 16. Wi-Fi watt hours.

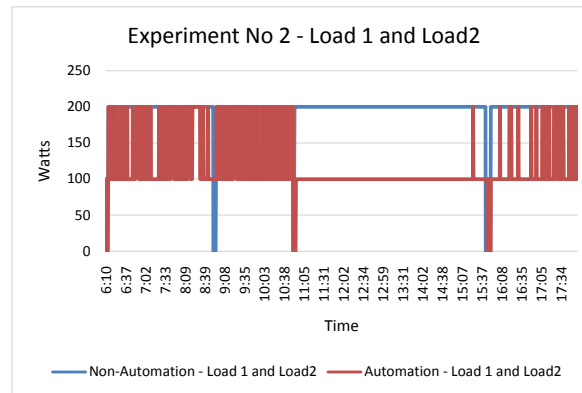


Figure 17. Load 1 and Load 2.

Table 10. Experiment 3—non-automation watts consumption.

Running Time	Total Wh	SH-HC-08 Wh	ESP8266 Wh
10 Mins	1.10	0.37	0.58
30 Mins	3.14	1.04	1.66
60 Mins	5.80	1.92	3.07
3 Hours	15.65	5.35	8.13
6 Hours	31.11	10.86	15.92
10.02 Hours	50.73	17.85	25.82

A table contains data fetched from loads are given in **Table 11**, as shown in the table both loads were switched on for 09 hours and 56 minutes out of 10 hours and 2 minutes, it was the longest time that any two of loads were switched on however the experiment was taken during the night and all of the loads were deliberately left switched on by sending commands through BLE to both non-automation and automation controller, however, only load 1 in automation were left to be switched on/off if any motion detected, both loads in non-automation controller were switched on for 99.23% of total duration of 10.02 hours, load 1 was remained on for 09:56:25 however load 2 was on for 09:56:44, non-automation controllers and their loads completely relies on user's commands to get send the load in to switch on/off mode by sending commands through mobile application connected via BLE.

Table 12 belongs to the automation controller, as this controller's main functionality is to keep load 1 to switch off mode if no resident is home. Automation controller consumed a total of 42.59 watts of energy during 10.02 hours of the running device time, which is 16.04% less consumption of energy. In the same way, look at the SH-HC-08 consumed 14.33 watts of energy that is 19.72% less consumption as compare to SH-HC-08 from the non-automation.

Table 13 contains the data that belongs to the loads from automation table. Load 1 was only switched on for 19 minutes and 30 seconds in total duration of

10.02 hours of running time of automation controller, which is 3.24% of the total running time of the controller. It was because only two occasion residents made several entries to the kitchen and rest of the time the resident was away from the kitchen which kept load 1 usage at the lowest level, however, load 2 in automation controller and both loads from non-automation controller were forced to stay switch on therefore those loads were switched on for maximum time with 99.15% of the total time they were on, in comparison with load 1 in automation based on motion detector was only 3.24% only which is 95.91% less. These benefits were because of the others loads were forced to stay on, however, load 1 based on motion detector got the advantage as resident didn't enter to the kitchen and stayed away most of the time.

Table 11. Experiment 3—non-automation loads.

Running Time	Load 1	Load 2
10 Mins	00:04:27	00:04:46
30 Mins	00:24:40	00:24:59
60 Mins	00:54:46	00:55:05
3 Hours	02:54:12	02:54:31
6 Hours	05:54:50	05:55:09
10.02 Hours	09:56:25	09:56:44

Table 12. Experiment 3—automation watts consumption.

Running Time	Total Wh	SH-SH-08 Wh	ESP8266 Wh
10 Mins	0.67	0.22	0.33
30 Mins	2.08	0.69	1.02
60 Mins	4.14	1.39	2.03
3 Hours	12.56	4.23	6.14
6 Hours	25.32	8.52	12.37
10.02 Hours	42.59	14.33	20.81

Table 13. Experiment 3—automation loads.

Running Time	Load 1	Load 2
10 Mins	00:03:17	00:03:11
30 Mins	00:03:17	00:23:24
60 Mins	00:03:17	00:53:30
3 Hours	00:10:15	02:52:56
6 Hours	00:10:15	05:53:34
10.02 Hours	00:19:30	09:55:09

5.3.1. Main Watts Hours

A comparison graph of automation and non-automation given, in which x-axis represents to the time and y-axis for watts, as shown in the graph difference is much and in percentage, the difference is 16.04% less than the non-automation controller. In the start of the graph for up to 20 minutes, there wasn't a big difference as shown in **Figure 18**; however, it was built up slowly and gradually up to 16.04% by the end of the graph. Non-automation board has consumed 50.72 watts of energy during the whole running time of the controller. However, automation controller consumed 42.58 watts of energy which is 8.14 watts less.

5.3.2. SH-HC-08 BLE Watts Hours

In the same way, a comparison graph of SH-HC-08 from both non-automation and automation controller are shown in **Figure 19**, BLE from automation controller consumed less energy as compare to another BLE from non-automation controller, it consumed 17.85 watts of energy in 10.02 hours, however BLE from automation controller just consumed 14.33 which 3.52 watts or 19.72% less. It has a tiny difference between both up to 1 hour of running time, however, the difference gradually increased up to 19.72%. it could have further reduced if it would have sent BLE as well in to switch off mode while no motion detected so no user would like to connect through BLE.

5.3.3. ESP8266 Wi-Fi Watts Hours

A comparison graph of the ESP8266 Wi-Fi watts hour is given in **Figure 20**, the graph shows that ESP8266 from non-automation controller consumed 25.82 watts in total duration, however, ESP8266 from automation consumed 20.81 watts which is 19.40% less, in automation ESP8266 consumption can also be pulled down by sending them into switch off mode while no resident home, but in future recommendation as willing to send and receive commands through web to control these controllers than ESP8266 could never be sent to switch off mode at the anytime users may want to control through web.

5.3.4. Load 1 & Load 2

It has explained earlier that the Experiment 3 conducted during the night, however, load 1, load 2 from non-automation controller and load 2 from automation controller were forced to stay on, however, load 1 from automation controller relied on motion detector and only 2 various occasions activity has been noticed and rest of the time only the load were switched on those were forced to do so which was switched on by sending BLE commands through mobile apps connected to via Bluetooth. In the non-automation controller, both loads were on 1349 occasion here loads from automation were remained switch on just only 35 occasions as shown in **Figure 21**.

5.3.5. Temperature by DHT11

The graph shown in **Figure 22** belongs to the automation controller. A DHT11 mounted on the board to get the current temperature from the area where au-

tomation controller working on. It is to only get the temperature and send readings to the controller. However, no task has been specified for the controller to accomplish at this time.

It is for only future recommendation to add new load 3 put on the controller so it can switch on the room temperature controller load which could be a fan and/or heater. It may help to control room temperature by setting up a threshold

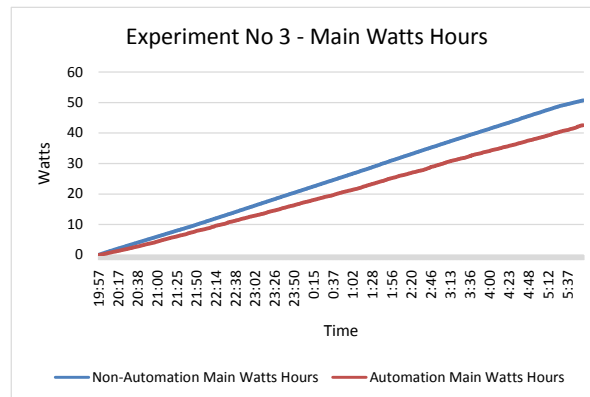


Figure 18. Main watt hours.

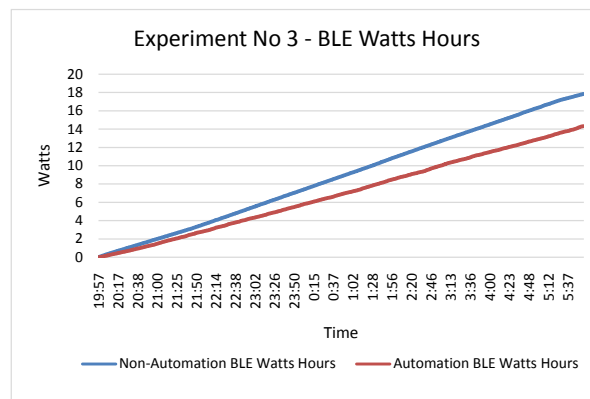


Figure 19. Bluetooth low energy watt hours.

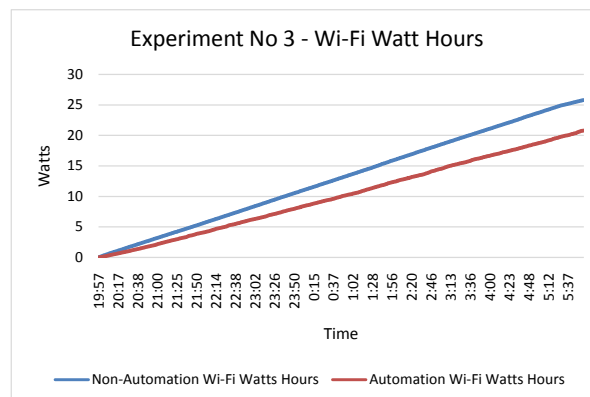


Figure 20. Wi-Fi watt hours.

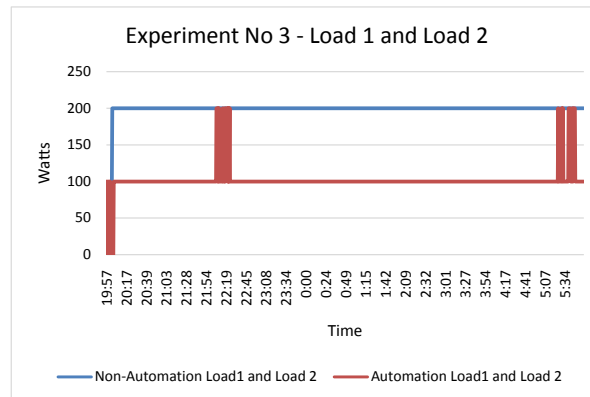


Figure 21. Total Load 1 and Load 2.

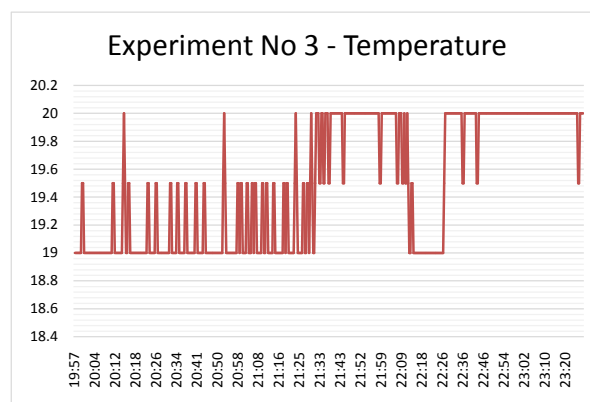


Figure 22. Temperature.

which triggers the fan or other cooling/heating devices on as per the threshold set up. The minimum temperature was recorded 19 however maximum 20 were recorded or mostly temperature stayed in between 19 to 20 degree Celsius.

5.4. Outcomes

In order to get better results, series of experiment conducted, however three experiments selected in and have shown individually with the results and outcome, as have explained earlier out of 2 loads only a Load 1 was controlled through motion detector sensor, while Load 2 was controlled through either scheduled by resident or directly switch on/off by sending command through mobile to controller. the total average of savings got here is most of the saved by Load 1, if Load 2 was also set with same rules as per Load 1 could have increase savings further, another table is given shows the total percentage of switch off time for the loads, represents savings of time in percentage.

In **Table 14** that shows a summary of all 3 experiments of loads from the non-automation controller, these loads were controlled through human intervention by sending commands through Bluetooth Low Energy, non-automation controller completely relies on human intervention. The loads could only be

switched on/off by sending commands by residents through mobile application, in 18 hours of total running time of non-automation controller average of both loads were on for 11 hours 54 minutes and 12 seconds, that is 66.13% of the running time loads were switched on, in other words, 33.87% of the running time loads were kept switch off by sending commands from residents. There is another **Table 15** shown that belongs to the automation controller.

In **Table 15**, Load 1 was kept the switched on 02:45:11 by automation controller however Load 2 which is either controlled by residents or scheduled by residents and kept the switch on for 11:45:46. The average running time of automation loads recorded 07:15:28 which is 40.32% of the total running time, as shown in the table there are 5 break points and switch on times for the loads were kept in between 35.78% to 40.32%, with the benefit of the proposed algorithm of Presence Aware of Power Saving Mode (PA-PSM) automation was able to kept loads switch off up to 59.68% of the total running time of controller and automation controller at least 25.81% more time loads were kept off than the non-automation controller's loads so that 25.81% is the actual savings have made by the developed artefact.

6. Conclusion

Researchers are continuously bringing new smart devices and paying highest attention toward the Internet of Things (IoT) to develop new devices by enabling Artificial Intelligence in, and it is to help digital system to interact with the real environment of the world without human intervention and/or minimizing human interaction at the lowest level. The computer researchers believe that a gigantic volume of interconnected smart devices produce a massive amount of data to

Table 14. Non-automation (Exp1 + Exp2 + Exp3).

Running Time	Average (Load 1 + Load 2)	% of Load ON time	% of Load OFF time
30 Mins/0.5Hour	00:14:06	46.97%	53.03%
90 Mins/1.5Hours	00:57:24	63.77%	36.23%
180 Mins/3Hours	01:57:53	65.50%	34.50%
540 Mins/9Hours	05:53:25	65.45%	34.55%
10,800 Mins/18Hours	11:54:12	66.13%	33.87%

Table 15. Automation (Exp1 + Exp2 + Exp3).

Running Time	Average (Load 1 + Load 2)	% of Load ON time	% of Load OFF time
30 Mins/0.5Hour	00:10:44	35.78%	64.22%
90 Mins/1.5Hours	00:35:05	38.98%	61.02%
180 Mins/3Hours	01:12:52	40.49%	59.51%
540 Mins/9Hours	03:46:14	41.90%	58.10%
10,800 Mins/18Hours	07:15:28	40.32%	59.68%

help to provide best services to human society [26]. Therefore, in [27] [28] researchers believe in the near future 50 - 200 bn interconnected smart devices will be able to serve human society by 2020. Moreover, these devices will be running by energy means; it will also increase huge energy consumption. Therefore, researchers need to pay attention to reducing the energy consumption of IoT devices as much as possible to reduce down energy cost and energy scarcity. Therefore, “**Presence Aware Power Saving Mode (PA-PSM) Enhancement for IoT Devices for Energy Conservation**”, has been proposed which help to reduce energy consumption by 25.81% in smart devices in smart home network. The proposed system utilizes BLE 5.0 to maintain the smart home network, in which sensors find human presence and keep devices in working mode nearby human and the rest of smart devices stay inactive or deep sleep mode. The non-automation and automation controllers built and conducted a series of experiments and results have suggested that 25.81% of the running time the loads were in switched off mode. This result has been achieved because of a proposed algorithm helped to reduce energy consumption in smart homes. A proposed algorithm, several sensors including BLE and Wi-Fi altogether makes Presence Aware Power Saving Mode (PA-PSM) enhancement for IoT devices for energy conservation, to reduce energy consumption in individual devices in smart homes. Sensors find human presence within home then sends information to controller and it sends devices to switch on mode else keeps switch-off mode to save energy, as in [27] [28], research suggested 50 - 200 bn of devices would be connected to IoT therefore, an average of approx. 7 devices per head as per 50 bn smart devices/7.7bn human population [29]. According to research, nearly 7 smart devices would be connected to the single smart home and the propose system may help to reduce nearly 25.81% of the energy consumed by smart devices within the smart home.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Sheng, Z.G., Yang, S., Yu, Y., Vasilakos, A.V., Mccann, J.A. and Leung, K.K. (2013) A Survey on the Ietf Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities. *IEEE Wireless Communications*, **20**, 91-98. <https://doi.org/10.1109/MWC.2013.6704479>
- [2] Lee, E.-J., Kim, C.-H. and Jung, I.Y. (2014) An Intelligent Green Service in the Internet of Things. *The Journal of Convergence*, **5**, 4-8.
- [3] Bae, S. (2014) Power Consumption Analysis of Prominent Time Synchronization Protocols for Wireless Sensor Networks. *Journal of Information Processing Systems*, **10**, 300-313. <https://doi.org/10.3745/JIPS.03.0006>
- [4] Perumal, T., Sulaiman, M.N., Mustapha, N., Shahi, A. and Thinaharan, R. (2014) Proactive Architecture for the Internet of Things (IoT) Management in Smart Homes. 2014 *IEEE 3rd Global Conference on Consumer Electronics (GCCE)*,

- Tokyo, 7-10 October 2014, 16-17. <https://doi.org/10.1109/GCCE.2014.7031347>
- [5] Lee, W., Cho, S., Chu, P., Vu, H., Helal, S., Song, W., Jeong, Y.-S. and Cho, K. (2016) Automatic Agent Generation for IoT-Based Smarthouse Simulator. *Neuro-computing*, **209**, 14-24. <https://doi.org/10.1016/j.neucom.2015.04.130>
 - [6] Jacobsson, A., Boldt, M. and Carlsson, B. (2016) A Risk Analysis of a Smart Home Automation System. *Future Generation Computer Systems*, **56**, 719-733.
 - [7] Fensel, A., Kumar, V. and Kathrin Tomic, S.D. (2014) End-User Interfaces for Energy-Efficient Semantically Enabled Smart Homes. *Energy Efficiency*, **7**, 655-675. <https://doi.org/10.1007/s12053-013-9246-2>
 - [8] Basso, G., Ferrez, P., Gabioud, D. and Roduit, P. (2015) An Extensible Simulator for Dynamic Control of Residential Area: A Case Study on Heating Control. 2015 *Euro-micro Conference on Digital System Design*, Funchal, Portugal, 26-28 August 2015, 486-493. <https://doi.org/10.1109/DSD.2015.76>
 - [9] Marnerides, A.K., Smith, P., Schaeffer-Filho, A. and Mauthe, A. (2015) Power Consumption Profiling Using Energy Time-Frequency Distributions in Smart Grids. *IEEE Communications Letters*, **19**, 46-49. <https://doi.org/10.1109/LCOMM.2014.2371035>
 - [10] Zhang, L.M., Liu, B.C., Tang, Q.H. and Wu, L.P. (2014) The Development and Technological Research of Intelligent Electrical Building. 2014 *China International Conference on Electricity Distribution*, Shenzhen, 23-26 September 2014, 88-92. <https://doi.org/10.1109/CICED.2014.6991669>
 - [11] Samadi, P., Mohsenian-Rad, H., Wong, V.W.S. and Schober, R. (2013) Tackling the Load Uncertainty Challenges for Energy Consumption Scheduling in Smart Grid. *IEEE Transactions on Smart Grid*, **4**, 1007-1016. <https://doi.org/10.1109/TSG.2012.2234769>
 - [12] Ulukus, S., et al. (2015) Energy Harvesting Wireless Communications: A Review of Recent Advances. *IEEE Journal on Selected Areas in Communications*, **33**, 360-381. <https://doi.org/10.1109/JSA.2015.2391531>
 - [13] Bluetooth SIG (2017) Bluetooth Specification Version 4.0. <https://www.bluetooth.com/specifications/adopted-specifications>
 - [14] IEEE Standard for Low-Rate Wireless Networks (2016) IEEE Standard for Low-Rate Wireless Networks—Amendment 2: Ultra-Low Power Physical Layer. IEEE Std 802.15.4q-2016, 1-52.
 - [15] IEEE Approved Draft Standard for IT (2013) IEEE Approved Draft Standard for IT-Telecommunications and Information Exchange between Systems-LAN/MAN-Specific Requirements. Part 11: Wireless LAN Medium Access Control and Physical Layer Specifications. IEEE P802.11ac/D7.0, 1-456.
 - [16] Langhammer, N. and Kays, R. (2012) Enhanced Frequency Hopping for Reliable Interconnection of Low Power Smart Home Devices. 2012 *8th International Wireless Communications and Mobile Computing Conference*, Limassol, Cyprus, 27-31 August 2012, 305-310. <https://doi.org/10.1109/IWCMC.2012.6314221>
 - [17] Alba, M., Avalos, M., Guzmán, C. and Larios, V.M. (2016) Synergy between Smart Cities' Hackathons and Living Labs as a Vehicle for Accelerating Tangible Innovations on Cities. 2016 *IEEE International Smart Cities Conference*, Trento, Italy, 12-15 September 2016, 1-6. <https://doi.org/10.1109/ISC2.2016.7580877>
 - [18] Bluetooth SIG (2017) How It Works. <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works>
 - [19] Nordic Semiconductor (2017) Bluetooth 5, Taking Bluetooth Further.

- <https://www.nordicsemi.com/eng/Products/Bluetooth-5>
- [20] MathWorks (2017) Simulink-Getting Started Guide.
https://uk.mathworks.com/help/pdf_doc/simulink/sl_gs.pdf
- [21] Autodesk (2019) PCB Made Easy, Autodesk.
<https://www.autodesk.com/products/eagle/overview>
- [22] Express PCB (2019) Resources for Engineers. <https://www.expresspcb.com/>
- [23] Labcenter, Proteus Labcenter (2019) Proteus in Education.
<https://www.labcenter.com/>
- [24] Yasin, S. (2018) Top 10 Best PCB Design Software Tools for Electronics Engineers by PCBgogo. Electronics Lovers.
<https://www.electronicshobby.com/2018/08/top-10-best-pcb-design-software-tools.html>
- [25] Amirtharajah, R. and Robatmili, B. (2016) Hot Chips 27. *IEEE Micro*, **36**, 6-7.
<https://doi.org/10.1109/MM.2016.31>
- [26] Familiar, B. (2015) Microservices, IoT, and Azure. In: Microservices, I.A., Ed., *Microservices, IoT, and Azure*, Apress, Berkeley, CA, 133-163.
https://doi.org/10.1007/978-1-4842-1275-2_7
- [27] Gray, C., Ayre, R., Hinton, K. and Tucker, R.S. (2015) Power Consumption of IoT Access Network Technologies. 2015 *IEEE International Conference on Communication Workshop*, London, 8-12 June 2015, 2818-2823.
<https://doi.org/10.1109/ICCW.2015.7247606>
- [28] CISCO (2016) Internet of Things (IoT).
<http://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html>
- [29] Worldmeters (2019) Current World Population.
<https://www.worldometers.info/world-population/>